

CLAIMS:

What is claimed is:

1. A apparatus comprising:
 - a first processor and a second processor;
 - a plurality of memory devices coupled to the first processor and the second processor;
 - a register buffer coupled to the first processor and the second processor;
 - a trace buffer coupled to the first processor and the second processor;and
 - a plurality of memory instruction buffers coupled to the first processor and the second processor;wherein the first processor and the second processor perform single threaded applications using multithreading resources.
2. The apparatus of claim 1, wherein the memory devices comprise of a plurality of cache devices.
3. The apparatus of claim 1, wherein the first processor is coupled to at least one of a plurality of zero level (L0) data cache devices and at least one of a plurality of L0 instruction cache devices, and the second processor is coupled to at least one of the plurality of L0 data cache devices and at least one of the plurality of L0 instruction cache devices.
4. The apparatus of claim 3, wherein each of the plurality of L0 data cache devices having exact copies of data cache instructions, and each of the plurality of L0 instruction cache devices having exact copies of instruction cache instructions.
5. The apparatus of claim 1, wherein the plurality of memory instruction buffers includes at least one store forwarding buffer and at least one load-ordering buffer.

6. The apparatus of claim 5, the at least one store forwarding buffer comprising a structure having a plurality of entries, each of the plurality of entries having a tag portion, a validity portion, a data portion, a store instruction identification (ID) portion, and a thread ID portion.

7. The apparatus of claim 6, the at least one load ordering buffer comprising a structure having a plurality of entries, each of the plurality of entries having a tag portion, an entry validity portion, a load identification (ID) portion, and a load thread ID portion.

8. The apparatus of claim 7, each of the plurality of entries further having a store thread ID portion, a store instruction ID portion, and a store instruction validity portion.

9. The apparatus of claim 1, the trace buffer is a circular buffer having an array with head and tail pointers, the head and tail pointers having a wrap-around bit.

10. The apparatus of claim 1, the register buffer comprising an integer register buffer and a predicate register buffer.

11. A method comprising:
executing a plurality of instructions in a first thread by a first processor;
and
executing the plurality of instructions in the first thread by a second processor as directed by the first processor, the second processor executing the plurality of instructions ahead of the first processor.

12. The method of claim 11, further including:
transmitting control flow information from the second processor to the first processor, the first processor avoiding branch prediction by receiving the control flow information; and
transmitting results from the second processor to the first processor, the first processor avoiding executing a portion of instructions by committing the results of the portion of instructions into a register file from a trace buffer.

13. The method of claim 12, further including:
duplicating memory information in separate memory devices for
independent access by the first processor and the second processor.

14. The method of claim 12, further including:
clearing a store validity bit and setting a mispredicted bit in a load entry
in the trace buffer if a replayed store instruction has a matching store
identification (ID) portion.

15. The method of claim 12, further including:
setting a store validity bit if a store instruction that is not replayed
matches a store identification (ID) portion.

16. The method of claim 12, further including:
flushing a pipeline, setting a mispredicted bit in a load entry in the trace
buffer and restarting a load instruction if one of the load is not replayed and
does not match a tag portion in a load buffer, and the load instruction matches
the tag portion in the load buffer while a store valid bit is not set.

17. The method of claim 12, further including:
executing a replay mode at a first instruction of a speculative thread;
terminating the replay mode and the execution of the speculative thread
if a partition in the trace buffer is approaching an empty state.

18. The method of claim 12, further including:
supplying names from the trace buffer to preclude register renaming;
issuing all instructions up to a next replayed instruction including
dependent instructions;
issuing instructions that are not replayed as no-operation (NOPS)
instructions;
issuing all load instructions and store instructions to memory;
committing non-replayed instructions from the trace buffer to the
register file.

19. The method of claim 12, further including:
clearing a valid bit in an entry in a load buffer if the load entry is retired.

20. An apparatus comprising a machine-readable medium containing instructions which, when executed by a machine, cause the machine to perform operations comprising:
executing a first thread from a first processor; and
executing the first thread from a second processor as directed by the first processor, the second processor executing instructions ahead of the first processor.

21. The apparatus of claim 20, further containing instructions which, when executed by a machine, cause the machine to perform operations including:
transmitting control flow information from the second processor to the first processor, the first processor avoiding branch prediction by receiving the control flow information.

22. The apparatus of claim 21, further containing instructions which, when executed by a machine, cause the machine to perform operations including:
duplicating memory information in separate memory devices for independent access by the first processor and the second processor.

23. The apparatus of claim 21, further containing instructions which, when executed by a machine, cause the machine to perform operations including:
clearing a store validity bit and setting a mispredicted bit in a load entry in the trace buffer if a replayed store instruction has a matching store identification (ID) portion.

24. The apparatus of claim 21, further containing instructions which, when executed by a machine, cause the machine to perform operations including:
setting a store validity bit if a store instruction that is not replayed matches a store identification (ID) portion.

25. The apparatus of claim 21, further containing instructions which, when executed by a machine, cause the machine to perform operations including:

flushing a pipeline, setting a mispredicted bit in a load entry in the trace buffer and restarting a load instruction if one of the load is not replayed and does not match a tag portion in a load buffer, and the load instruction matches the tag portion in the load buffer while a store valid bit is not set.

26. The apparatus of claim 21, further containing instructions which, when executed by a machine, cause the machine to perform operations including:

executing a replay mode at a first instruction of a speculative thread;

terminating the replay mode and the execution of the speculative thread if a partition in the trace buffer is approaching an empty state.

27. The apparatus of claim 21, further containing instructions which, when executed by a machine, cause the machine to perform operations including:

supplying names from the trace buffer to preclude register renaming;

issuing all instructions up to a next replayed instruction including dependent instructions;

issuing instructions that are not replayed as no-operation (NOPS) instructions;

issuing all load instructions and store instructions to memory;

committing non-replayed instructions from the trace buffer to the register file.

28. The apparatus of claim 21, further containing instructions which, when executed by a machine, cause the machine to perform operations including:

clearing a valid bit in an entry in a load buffer if the load entry is retired.

29. A system comprising:

a first processor;

a second processor;

a bus coupled to the first processor and the second processor;

a main memory coupled to the bus;

a plurality of local memory devices coupled to the first processor and the second processor;

a register buffer coupled to the first processor and the second processor;

and

a trace buffer coupled to the first processor and the second processor;

a plurality of memory instruction buffers coupled to the first processor and the second processor,

wherein the first processor and the second processor perform single threaded applications using multithreading resources.

30. The system of claim 29, the local memory devices comprise a plurality of cache devices.

31. The system of claim 30, the first processor is coupled to at least one of a plurality of zero level (L0) data cache devices and at least one of a plurality of L0 instruction cache devices, and the second processor is coupled to at least one of the plurality of L0 data cache devices and at least one of the plurality of L0 instruction cache devices.

32. The system of claim 31, wherein each of the plurality of L0 data cache devices having exact copies of data cache instructions, and each of the plurality of L0 instruction cache devices having exact copies of instruction cache instructions.

33. The system of claim 31, the first processor and the second processor each sharing a first level (L1) cache device and a second level (L2) cache device.

34. The system of claim 29, wherein the plurality of memory instruction buffers includes at least one store forwarding buffer and at least one load ordering buffer.

35. The system of claim 34, the at least one store forwarding buffer including a structure having a plurality of entries, each of the plurality of entries having a tag portion, a validity portion, a data portion, a store instruction identification (ID) portion, and a thread ID portion.